

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of:

Toru NODA

Application No.:

Group Art Unit:

Filed: August 26, 2003

Examiner:

For: WEB SERVER HAVING FUNCTION OF JAVA SERVLET, METHOD FOR UPDATING
JAVA PROGRAM AND COMPUTER PROGRAM

**SUBMISSION OF CERTIFIED COPY OF PRIOR FOREIGN
APPLICATION IN ACCORDANCE
WITH THE REQUIREMENTS OF 37 C.F.R. § 1.55**

Commissioner for Patents
PO Box 1450
Alexandria, VA 22313-1450

Sir:

In accordance with the provisions of 37 C.F.R. § 1.55, the applicant(s) submit(s)
herewith a certified copy of the following foreign application:

Japanese Patent Application No(s). 2002-249266

Filed: August 28, 2002

It is respectfully requested that the applicant(s) be given the benefit of the foreign filing
date(s) as evidenced by the certified papers attached hereto, in accordance with the
requirements of 35 U.S.C. § 119.

Respectfully submitted,

STAAS & HALSEY LLP

Date: August 26, 2003

By: 

H. J. Staas

Registration No. 22,010

1201 New York Ave, N.W., Suite 700
Washington, D.C. 20005
Telephone: (202) 434-1500
Facsimile: (202) 434-1501

日本国特許庁
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出願年月日
Date of Application: 2002年 8月28日

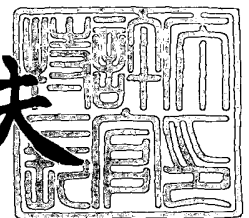
出願番号
Application Number: 特願2002-249266
[ST. 10/C]: [JP2002-249266]

出願人
Applicant(s): 富士通株式会社

2003年 7月24日

特許庁長官
Commissioner,
Japan Patent Office

今井康夫



出証番号 出証特2003-3058856

【書類名】 特許願

【整理番号】 0295450

【提出日】 平成14年 8月28日

【あて先】 特許庁長官殿

【国際特許分類】 G06F 9/06

【発明の名称】 J a v a サーブレットの機能を有する W e b サーバ、 J
a v a プログラムの更新方法、およびコンピュータプロ
グラム

【請求項の数】 5

【発明者】

【住所又は居所】 大阪府大阪市中心区城見二丁目 2 番 5 3 号 株式会社富
士通関西システムズ内

【氏名】 野田 徹

【特許出願人】

【識別番号】 000005223

【氏名又は名称】 富士通株式会社

【代理人】

【識別番号】 100086933

【弁理士】

【氏名又は名称】 久保 幸雄

【電話番号】 06-6304-1590

【手数料の表示】

【予納台帳番号】 010995

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 9704487

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 J a v a サブレットの機能を有する W e b サーバ、 J a v a プログラムの更新方法、およびコンピュータプログラム

【特許請求の範囲】

【請求項 1】

ネットワークユーザの端末装置に表示するための W e b ページの H T M L ファイルを生成する処理を行う J a v a プログラムを記憶する、 J a v a サブレットの機能を有する W e b サーバであって、

W e b ページの H T M L ファイルを当該 W e b ページの作成または修正を行った者の端末装置から取得する取得手段と、

取得された W e b ページの H T M L ファイルを J a v a のコード群に変換する第一の変換手段と、

取得された W e b ページの H T M L ファイルの、変換された J a v a のコード群を、前記 J a v a プログラムのうちの当該 W e b ページの構成に係る部分と置き換えることによって、当該 J a v a プログラムを更新するプログラム更新手段と、

指定された W e b ページの構成に係る部分の J a v a のコード群を前記 J a v a プログラムの中から抽出する抽出手段と、

抽出された J a v a のコード群を H T M L ファイルに変換する第二の変換手段と、

変換された H T M L ファイルを指定元の端末装置に送信する送信手段と、が設けられており、

前記第一の変換手段は、取得された H T M L ファイルに修正不可の部分を示すタグが含まれている場合に、当該タグを、修正不可の部分を示す J a v a のコードに変換し、

前記第二の変換手段は、抽出された J a v a のコード群に含まれる前記コードを前記タグに変換する、

ことを特徴とする J a v a サブレットの機能を有する W e b サーバ。

【請求項 2】

前記 J a v a プログラムは、1つの W e b ページについて演算の処理を行う業務ロジックおよび当該 W e b ページの構成に係る部分の H T M L のソースコードを生成する画面生成ロジックからなり、

前記第一の変換手段は、取得された H T M L ファイルに含まれる前記タグを、前記業務ロジックによる演算の結果を受け取る機能を有する J a v a のコードに変換する、

請求項 1 記載の J a v a サーブレットの機能を有する W e b サーバ。

【請求項 3】

前記取得手段によって取得された H T M L ファイルに他のファイルのパスが含まれる場合に、当該パスのパス名をパス情報として記憶するパス記憶手段と、

前記パス名を更新するパス更新手段と、が設けられており、

前記第二の変換手段は、抽出された J a v a のコード群のうちの他のファイルのパスを示す部分については、前記パス記憶手段に記憶されている前記パス情報に基づいて H T M L に変換する、

請求項 1 または請求項 2 記載の J a v a サーブレットの機能を有する W e b サーバ。

【請求項 4】

ネットワークユーザの端末装置に表示するための W e b ページの H T M L ファイルを生成する処理を行う J a v a プログラムを記憶する、 J a v a サーブレットの機能を有する W e b サーバにおける J a v a プログラムの更新方法であって、

指定された W e b ページの構成に係る部分の J a v a のコード群を前記 J a v a プログラムの中から抽出するステップと、

抽出された J a v a のコード群を H T M L ファイルに変換するステップと、

変換された H T M L ファイルを指定元の端末装置に送信するステップと、

前記指定元の端末装置から修正された H T M L ファイルを取得するステップと、

取得された H T M L ファイルを J a v a のコード群に変換するステップと、

変換された J a v a のコード群を、前記 J a v a プログラムのうちの当該コー

ド群に対応する W e b ページの構成に係る部分と置き換えることによって、当該 J a v a プログラムを更新するステップと、からなり、

前記 J a v a のコード群に変換するステップにおいて、取得された H T M L ファイルに修正不可の部分を示すタグが含まれている場合に、当該タグを、修正不可の部分を示す J a v a のコードに変換し、

前記 H T M L ファイルに変換するステップにおいて、抽出された J a v a のコード群に含まれる前記コードを前記タグに変換する、

ことを特徴とする J a v a プログラムの更新方法。

【請求項 5】

ネットワークユーザの端末装置に表示するための W e b ページの H T M L ファイルを生成する処理を行う H T M L 生成プログラムを記憶する、 J a v a サーブレットの機能を有するコンピュータに用いられるコンピュータプログラムであって、

W e b ページの H T M L ファイルを当該 W e b ページの作成または修正を行った者の端末装置から取得する処理と、

取得された W e b ページの H T M L ファイルを J a v a のコード群に変換する処理と、

取得された W e b ページの H T M L ファイルの、変換された J a v a のコード群を、前記 H T M L 生成プログラムのうちの当該 W e b ページの構成に係る部分と置き換えることによって、当該 H T M L 生成プログラムを更新するプログラム処理と、

指定された W e b ページの構成に係る部分の J a v a のコード群を前記 H T M L 生成プログラムの中から抽出する処理と、

抽出された J a v a のコード群を H T M L ファイルに変換する処理と、

変換された H T M L ファイルを指定元の端末装置に送信する処理と、を実行させ、

前記 J a v a のコード群に変換する処理において、取得された H T M L ファイルに修正不可の部分を示すタグが含まれている場合に、当該タグを、修正不可の部分を示す J a v a のコードに変換し、

前記HTMLファイルに変換する処理において、抽出されたJavaのコード群に含まれる前記コードを前記タグに変換する、

ようにコンピュータに実行させるためのコンピュータプログラム。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、Javaサーバレットの機能を有するWebサーバにおけるJavaプログラムの変更に関する。

【0002】

【従来の技術】

サン・マイクロシステムズ社のプログラム言語であるJava（米国 Sun Microsystems, Inc. の登録商標）技術の1つとして、Webサーバで用いられるJavaサーバレットがある。Javaサーバレットによってプログラムを実行すると、WWWブラウザからの要求に応じてWebページを表示するための記述言語であるHTML（Hypertext Markup Language）のハイパーテキストすなわちHTMLファイルを生成し、そのWWWブラウザに送信することができる。

【0003】

係るプログラムは、主に業務ロジック、画面生成（描画）ロジック、および制御ロジックからなる。業務ロジックは、ユーザ認証などの業務的な処理を行う。画面生成ロジックは、Webページのデザインなど構成に関するものであり、業務ロジックによる処理の結果に基づいてHTMLファイルを生成する。よって、係るプログラムを実行することにより、それぞれのユーザの指示などに応じた内容のWebページを表示するためのHTMLファイルを生成することができる。つまり、内容が固定的ではない、動的なWebページを各ユーザに提供することができる。

【0004】

このような特徴より、Javaサーバレットを利用してネットワークのユーザに対してWebによるサービス（例えばオンラインショッピングまたは情報検索など）のサイトを開設する業者も多い。

【 0 0 0 5 】

これらの業者は、J a v a サーブレットを利用したサイトを開設する際に、業務的な処理（例えばユーザ認証、商品の選択、課金、または検索など）を行う仕組みを構築し、さらにW e b ページのデザインを行わなければならない。これらの作業は大変なので、J a v a プログラミングのできるプログラマに業務ロジックおよび制御ロジックの作成の作業を行わせ、W e b デザイナに画面生成ロジックの作成の作業を行わせる場合がほとんどである。

【 0 0 0 6 】

プログラマまたはW e b デザイナは、サーブレットエンジン上で動作させるために、これらのロジックをJ a v a のコードまたはJ S P（Java Server Pages）によって作成しなければならない。

【 0 0 0 7 】

J a v a による作業を効率的に進めるために、例えば、特開 2 0 0 1 - 3 4 4 1 0 5 号公報に記載されるように、各画面とサーブレットとを 1 対 1 の関係で定義できるようにする技術が提案されている。

【 0 0 0 8 】**【発明が解決しようとする課題】**

しかし、J a v a または J S P による作業は、高度なスキルが要求されるので、上記の公報に記載の技術を用いたとしても、W e b デザイナにとって画面生成ロジックの作成および修正の作業は非常に難しい。

【 0 0 0 9 】

したがって、サイトの開設後、サイト内のW e b ページの更新の作業に手間が掛かってW e b ページの更新が遅れがちになり、またはプログラムの中にバグが生じやすくなるなど、サイトの円滑な運営を妨げてしまうおそれがあった。

【 0 0 1 0 】

本発明は、このような問題点に鑑み、J a v a サーブレットを用いたW e b サーバがある環境において、J a v a のスキルのないW e b デザイナであっても容易にW e b ページのデザインの変更を行うことができるようにすることを目的とする。

【0011】

【課題を解決するための手段】

本発明に係るWebサーバは、ネットワークユーザの端末装置に表示するためのWebページのHTMLファイルを生成する処理を行うJavaプログラムを記憶する、Javaサーバレットの機能を有するWebサーバであって、WebページのHTMLファイルを当該Webページの作成または修正を行った者の端末装置から取得する取得手段と、取得されたWebページのHTMLファイルをJavaのコード群に変換する第一の変換手段と、取得されたWebページのHTMLファイルの、変換されたJavaのコード群を、前記Javaプログラムの中の当該Webページの構成に係る部分と置き換えることによって、当該Javaプログラムを更新するプログラム更新手段と、指定されたWebページの構成に係る部分のJavaのコード群を前記Javaプログラムの中から抽出する抽出手段と、抽出されたJavaのコード群をHTMLファイルに変換する第二の変換手段と、変換されたHTMLファイルを指定元の端末装置に送信する送信手段と、が設けられており、前記第一の変換手段は、取得されたHTMLファイルに修正不可の部分を示すタグが含まれている場合に、当該タグを、修正不可の部分を示すJavaのコードに変換し、前記第二の変換手段は、抽出されたJavaのコード群に含まれる前記コードを前記タグに変換する。

【0012】

好ましくは、前記Javaプログラムは、1つのWebページについて演算の処理を行う業務ロジックおよび当該Webページの構成に係る部分のHTMLのソースコードを生成する画面生成ロジックからなり、前記第一の変換手段は、取得されたHTMLファイルに含まれる前記タグを、前記業務ロジックによる演算の結果を受け取る機能を有するJavaのコードに変換する。

【0013】

または、前記取得手段によって取得されたHTMLファイルに他のファイルのパスが含まれる場合に、当該パスのパス名をパス情報として記憶するパス記憶手段と、前記パス名を更新するパス更新手段と、が設けられており、前記第二の変換手段は、抽出されたJavaのコード群の中の他のファイルのパスを示す部

分については、前記パス記憶手段に記憶されている前記パス情報に基づいて HTML に変換する。

【0014】

本発明において、「Java のコード群」とは、Java のソースコードまたはこのソースコードをコンパイルして得られる Java バイトコード（中間コード）を示す。

【0015】

【発明の実施の形態】

図1は本発明に係るWebサーバ1が設けられたネットワークシステムの例を示す図、図2はWebサーバ1のハードウェア構成の例を示す図、図3はWebサーバ1のプログラムの例を示す図、図4はログイン画面HG1の例を示す図、図5はHTMLのソースコードの例を示す図、図6はJavaソースコードの例を示す図、図7は初期状態のWebページ編集画面HG2の例を示す図、図8はHTMLのソースコードが表示されたWebページ編集画面HG2の例を示す図、図9はWebサーバ1の機能的構成の例を示す図、図10は編集後のログイン画面HG1'を示す図、図11は更新が完了した旨のメッセージの画面を示す図である。

【0016】

本発明に係るWebサーバ1は、図1に示すように、ネットワーク4を介して端末装置2、3と接続されている。ネットワーク4として、インターネット、専用線、または公衆回線などが用いられる。

【0017】

端末装置3は、一般のユーザが所有するパーソナルコンピュータ、ワークステーション、PDA (Personal Digital Assistant)、または携帯電話端末などである。端末装置3にはWWWブラウザ (Web ブラウザ) の機能が備えられており、Webサーバ1などからHTML (Hypertext Markup Language) ファイルを取得することによって、種々の情報が記載されたWebページを閲覧し、または情報検索、オンラインショッピング、オークション、またはオンラインバンキングなどの種々のサービスのWebアプリケーションを利用することができる。

【0018】

端末装置2は、これらのサービスの提供の際に用いられるWebページを作成しまたは編集するためのものである。例えば、Webページに季節に応じた挿し絵を入れたり、誤字が見つかった場合に修正したりする際に用いられる。端末装置2として、例えばWWWブラウザがインストールされているパーソナルコンピュータまたはワークステーションなどが用いられる。

【0019】

Webサーバ1は、図2に示すように、CPU10a、RAM10b、ROM10c、磁気記憶装置10d、ディスプレイ装置10e、およびマウスまたはキーボードなどの入力装置10fなどによって構成される。

【0020】

磁気記憶装置10dには、オペレーティングシステム(OS)のほか、図3に示すように、JavaサーブレットによるWebアプリケーションを実現するためのプログラムが記憶されている。これらのプログラムは、必要に応じてRAM10bにロードされ、CPU10aによって実行される。

【0021】

サーブレットエンジン71は、JavaサーブレットにおけるJava仮想マシン(JVM)に相当するものである。なお、サーブレットエンジンは、サーブレットAPI2.2以降では、サーブレットコンテナと呼称される。

【0022】

Javaソースジェネレータ72は、HTMLで記述されたソースコードに基づいてJavaソースコードを生成する、HTML-Java変換ツールである。Javaコンパイラ73は、Javaソースコードをサーブレットエンジン71で実行可能なJavaバイトコードであるJavaクラス(クラスファイル)にコンパイルする。

【0023】

Webアプリケーション80は、端末装置3に対してサービスを提供するためのプログラムであり、複数のJavaクラス(サーブレットクラス)によって構成される。これらの中には、例えば図4に示すログイン画面HG1のような画面

(Web ページ) ごとの J a v a クラスが含まれている。このような画面の J a v a クラスとして、画面生成 (描画) ロジックを実現するための画面生成ロジッククラスおよび業務ロジックを実現するための業務ロジッククラスがある。基本的に、1つの画面について画面生成ロジッククラスおよび業務ロジッククラスが少なくとも1つずつ用意されている。または、画面生成ロジックおよび業務ロジックが1つの J a v a クラスに含まれるように構成してもよい。

【0024】

業務ロジックは、端末装置 3 における操作などに基づいて所定の演算処理を行う。例えば、ログイン画面 H G 1 が表示された端末装置 3 において入力されたユーザ I D およびパスワードに基づいてユーザ認証を行う。画面生成ロジックは、画面 (Web ページ) の H T M L ファイルを生成する処理を行う。業務ロジックと画面生成ロジックとは処理の流れが連携されており、例えば、画面生成ロジックは業務ロジックの処理結果 (実行結果) に基づいて H T M L ファイルを生成する。

【0025】

画面生成ロジックは、次のような手順で作成される。まず、H T M L で図 5 に示すように画面生成ロジックの基となる部分をコーディングする。コーディングされたソースコードを J a v a ソースジェネレータ 7 2 によって図 6 に示すように J a v a ソースコードに変換する。そして、これを J a v a コンパイラ 7 3 によって J a v a バイトコードにコンパイルし、画面生成ロジックが得られる。

【0026】

H T M L のソースコードには、この H T M L のソースコードを編集 (メンテナンス) する際に書き換えてはならない個所を示す本発明のオリジナルのタグを用いることができる。以下、係るタグを「X タグ」を呼称する。

【0027】

X タグは、「< X n a m e = " … " > … < / X >」または「< X = … = X >」のように用いられる。例えば、図 4 のログイン画面 H G 1 において、通常ユーザ I D のテキストボックスの既定値 (v a l u e = … に示される値) として変数「U S E R _ I D」に格納されている値を用いる場合は「< I N P U T t y p

e="text" name="USER_ID" value="USER_ID">」のようにコードを記述するが、この既定値の部分を編集不可であるものとして表したい場合は「<INPUT type="text" name="USER_ID" value="<X=USER_ID=X>>」のように記述すればよい。このように、Xタグは、HTMLのソースのうちの動的部分に対して用いることができる。「動的部分」とは、画面のデザイン（文字または画像など）のうち業務ロジックの処理の結果に応じて表示の内容が異なる部分という。Xタグに対する処理については後に説明する。

【0028】

一方、業務ロジックは、プログラマによってコーディングされたJavaソースコードをJavaコンパイラ73でコンパイルすることによって得られる。

【0029】

また、磁気記憶装置10dには、図3に示すように、画面生成ロジックをWebデザイナの端末装置2において編集（メンテナンス）するための編集アプリケーション74も記憶されている。編集アプリケーション74によると、例えば、端末装置2において図7または図8に示すWebページ編集画面HG2によってWebページ（HTMLファイル）をメンテナンスすることができる。編集アプリケーション74も、画面生成ロジックおよび業務ロジックなどによって構成され、Webサーバ1のサーバレットエンジン上で動作する。

【0030】

このような構成により、Webサーバ1には、図9に示すように、HTMLファイル受信部101、指定受付部102、指定画面抽出部103、第一の変換部104、コード更新部105、第二の変換部106、HTMLファイル送信部107、パス記憶部108、およびパス更新部109などの機能が実現される。これらの機能によって、各ユーザに対して提供するサービスのためのWebページ（例えば図4のログイン画面HG1など）のデザインを変更することができる。以下、これらの機能について説明する。

【0031】

端末装置2のWWWブラウザにおいて、所定のURLが指定されると、Web

サーバ 1 から取得した HTML ファイルに基づいて図 7 に示す Web ページ編集画面 HG 2 が表示される。ここで、Web デザイナは、編集したい Web ページの ID（以下、「GID」と記載する。）、Web デザイナのユーザ ID およびパスワードをテキストボックス BX 2 1 ～ BX 2 3 にそれぞれ入力し、編集ボタン BN 2 1 をクリックする。例えば、図 4 に示すログイン画面 HG 1 を編集したい場合は、GID として「GID 0 1」を入力する。編集ボタン BN 2 1 がクリックされると、入力された GID、ユーザ ID、およびパスワードが Web サーバ 1 に対して送信される。

【 0 0 3 2 】

図 9 の指定受付部 1 0 2 は、端末装置 2 から取得したユーザ ID およびパスワードに基づいて端末装置 2 のユーザ（Web デザイナ）が正しいユーザであるか否かの認証つまりセキュリティのチェックを行う。そして、正しいユーザである旨の認証が得られた場合に、GID の指定を受理し、指定画面抽出部 1 0 3 に対してその GID を伝える。

【 0 0 3 3 】

指定画面抽出部 1 0 3 は、伝えられた GID に対応する画面生成ロジックの Java バイトコードを磁気記憶装置 1 0 d より抽出する。第二の変換部 1 0 6 は、抽出された Java バイトコードを HTML のソースコードに変換する。HTML ファイル送信部 1 0 7 は、変換された HTML のソースコードを編集する画面を表示するための HTML ファイルを端末装置 2 に送信する。

【 0 0 3 4 】

このとき、端末装置 2 において、Web ページ編集画面 HG 2 のテキストボックス BX 2 4 には、図 8 に示すように、Web サーバ 1 から受信した HTML のソースコードが表示される。Web デザイナは、この HTML のソースコードを修正することによって、Web ページのデザインの編集を行うことができる。レビューボタン BN 2 3 がクリックされると、修正された HTML のソースコードに基づいて WWW ブラウザに図 1 0 に示すようなログイン画面 HG 1' が表示（レビュー）される。Web デザイナは、これを見て編集の出来具合を確認することができる。そして、Web ページ編集画面 HG 2 に戻ってアップデート

ボタンBN22がクリックされると、修正されたHTMLのソースコードがWebサーバ1に対して送信される。

【0035】

Webサーバ1において、修正されたHTMLのソースコードを図9のHTMLファイル受信部101が受信すると、第一の変換部104は、このHTMLのソースコードをJavaソースコードに変換し、さらにJavaバイトコードに変換（コンパイル）する。

【0036】

コード更新部105は、変換されたJavaバイトコードを元のJavaバイトコードに上書きすることによって、画面生成ロジックのコードを更新（差替え）する。このとき、端末装置3には、図11に示すような画面生成ロジッククラスの更新が完了した旨のメッセージが表示される。

【0037】

図12は画面HG4（Webページ）の例を示す図、図13は図12の画面HG4のためのHTMLのソースコードの例を示す図、図14は業務ロジックおよび画面生成ロジックにおけるXタグの機能を説明する図、図15は図13（a）のHTMLのソースコードを編集した例を示す図である。

【0038】

次に、HTMLのXタグの処理とともに、HTMLとJavaとの間でのコード変換について説明する。前に述べた通り、Webデザイナーは、HTMLによってWebページの作成およびメンテナンスを行う。例えば、図12のWebページ（画面HG4）を作成する場合は、図13（a）のようにHTMLのソースコードを記述する。画面HG4の「大阪府」という文字列の部分を動的部分として扱いたい場合は、Xタグを用いる。すなわち、図13（b）のように「大阪府」の部分をXタグでマーキングすればよい。

【0039】

図9の第一の変換部104は、HTMLのソースコードを、次のようなルールでJavaソースコードに変換する。

（a） Xタグでマーキングされていない部分は、HTMLのソースコードに示

されるそのままの画面を表示させるように機能するコードに変換する。

(b) Xタグでマーキングされている部分は、このHTMLのソースコードを基に生成された画面生成ロジックの実行時に、業務ロジックからマーキング名 (name=…が示す値) に基づいて値を受け取るように機能する「getHTML」というコードに変換する (図5、図6参照)。例えば、図13 (b) のソースコードに含まれる「<X name=" ADDRESS" >大阪府</X>」の場合は、図14に示すように業務ロジックの実行結果を「ADDRESS」という名の変数を介して取得するように機能するコードに変換する。

(c) 画像のURL (パス) またはサブミット先のURLとなる部分は、そのURLを外部ファイルから動的に読み取ることができるように機能するコードに変換する。つまり、例えば、次のようにしてコード化を行う。図9のパス記憶部108は、端末装置2から得たHTMLのソースコードに含まれる画像などのURLを抽出し、そのうちのファイル名以外の部分すなわちパス名をパス情報89として記憶する。例えば、HTMLのソースコードに「src="/test1/images/gazou.gif"」という記述があった場合は、「/test1/images/」をパス情報89として外部ファイルに記憶する。第一の変換部104は、HTMLのソースコードに含まれるURLのうちのパス名の部分を、パス情報89を識別する文字列と置き換えた後、Javaソースコードに変換する。

【0040】

これらのルールに基づいて変換されたJavaソースコードは、コンパイルされてJavaバイトコードとなり、Javaサーブレットの実行環境下に格納される。以上のようにして、画像生成ロジックのHTMLのソースコードがJavaバイトコードに変換される。

【0041】

なお、前に述べた通り、プログラマは、業務ロジックについてはJavaのソースコードとして記述しなければならない。さらに、画面生成ロジックと連携する業務ロジックについては、その画面生成ロジックとの間でデータをやり取りが行われるようコーディングしなければならない。例えば、図14の場合は、「A

DDRESS」によって値が出力（セット）される機能を業務ロジックに実装させなければならない。

【0042】

一方、図9の第二の変換部106は、画像生成ロジックのJavaバイトコードに「getHTML」に相当するコードが含まれている場合に、係る部分がXタグになるようにHTMLのソースコードに変換する。

【0043】

また、画像などのURL（パス）に相当する部分については、パス情報89に基づいてHTMLへの変換を行う。したがって、端末装置2から取得したHTMLのソースコードに示されるURLがそのまま復元されるとは限らず、異なるURLに変換されて得られる場合がある。

【0044】

例えば、テスト段階において端末装置2から取得したHTMLのソースコードに「/test1/images/gazou.gif」という画像のURLが含まれていたとする。このとき、パス情報89として「/test1/images/」というパス名が得られる。そして、本番環境に移行する際にシステム管理者の操作に従って図9のパス更新部109がそのパス情報89を「/honban/images/」と書き換えたとする。すると、JavaバイトコードをHTMLのソースコードに変換した際に、画像のURLは「/honban/images/gazou.gif」となる。このように、外部ファイルに含まれるパス情報89を変更可能とすることによって、同一の階層に多数の画像のURLが含まれる場合であっても、容易に動的に本番環境または次のテスト段階のURLへの変更を行うことができる。

【0045】

このようにして得られたHTMLのソースコードは、HTMLファイル送信部107によって端末装置2に送信され、Webデザイナーによって例えば図15のように編集される。編集されたソースコードは、再び第一の変換部104によってコンパイルなどが施されてJavaバイトコードとなり、そして元の画面生成ロジックと置き換えられる。

【 0 0 4 6 】

図 1 6 は W e b サーバ 1 における処理の流れを説明するフローチャート、図 1 7 は画面 H G 5 (W e b ページ) の例を示す図である。次に、端末装置 2 または端末装置 3 からアクセスがあった場合の W e b サーバ 1 の処理の流れを、図 1 6 に示すフローチャートを参照して説明する。

【 0 0 4 7 】

端末装置 2 から W e b ページのデザインの編集の指示があった場合は (# 1 1 で Y e s) 、指定された画面 I D (G I D) に対応する画面生成ロジックの J a v a バイトコードを抽出し、これに基づいて「 g e t H T M L 」の部分で X タグでマーキングした H T M L ファイルを生成し (# 1 2) 、図 8 に示すような W e b ページ編集画面 H G 2 のための H T M L ファイルを端末装置 2 に送信する (# 1 3) 。

【 0 0 4 8 】

W e b ページ編集画面 H G 2 が表示された端末装置 2 において、 W e b デザイナは、 X タグの部分を書き換えないように注意しながらテキストボックス B X 2 4 内の H T M L のソースコードを修正 (編集) する (# 1 4) 。例えば、図 1 3 (b) のソースコードのうちの X タグで示された動的部分以外の部分つまりデザインが固定的または静的な部分を編集し、図 1 5 のようなソースコードにする。編集されたソースコードは W e b サーバ 1 に送信され (# 1 5) 、 J a v a ソースコードに変換され (# 1 6) 、さらに J a v a バイトコードにコンパイルされる (# 1 7) 。そして、元の J a v a バイトコードと置き換えられ、これにより W e b ページのデザインの編集が完了する。編集の完了後、念のために、ログインの認証機能が問題なく作動するか否かを確認する。

【 0 0 4 9 】

一方、端末装置 3 からアクセスがあった場合は (# 1 1 で N o) 、まず、業務ロジックが実行される (# 2 1) 。次に、業務ロジックによる実行結果に基づいて画面生成ロジックが動作し、 H T M L ファイルが生成される (# 2 2) 。例えば、業務ロジックの実行の結果が「大阪市中心区域見」であれば、図 1 7 に示すように動的部分 M V が「大阪市中心区域見」となる画面 H G 5 が表示されるよう

な HTML ファイルが生成される。そして、生成された HTML ファイルが端末装置 3 に送信される（# 2 3）。

【 0 0 5 0 】

ステップ # 1 2 ～ # 1 8 および # 2 1 ～ # 2 3 の処理は、端末装置 2、3 からアクセスがあるごとに適宜繰り返される。

【 0 0 5 1 】

本実施形態によると、J a v a サブレットによって提供される W e b アプリケーションの画面のデザインの編集を、J a v a のスキルのない者であっても容易に行うことができる。また、編集は WWW ブラウザ上で行うことができるので、端末装置 2 には、F T P（File Transfer Protocol）のアプリケーションやコンパイラなどを用意しておく必要がない。

【 0 0 5 2 】

X タグを用いることにより、画面生成ロジックと業務ロジックとの連携を崩すようなバグを防止することができる。さらに、HTML のソースコードに含まれる画像ファイルの URL をパス情報として外部ファイルに記憶しておくことにより、テスト環境から本番環境への移行の際の URL の付替えの手間を少なくすることができる。

【 0 0 5 3 】

1 つの HTML のソースコードの中に複数個の X タグを用いることも可能である。第二の変換部 1 0 6 は、編集対象の W e b ページの HTML のソースコードを、J a v a バイトソースを変換して生成したが、J a v a ソースコードを変換して生成してもよい。

【 0 0 5 4 】

W e b デザイナは、HTML のソースコードの修正を、W e b ページ作成用アプリケーション（いわゆるホームページ作成ソフト）を用いて行ってもよい。ホームページ作成ソフトに、X タグの部分に修正禁止を示す特殊な印を付ける機能を設けてもよい。

【 0 0 5 5 】

その他、W e b サーバ 1、端末装置 2、3 の全体または各部の構成、プログラ

ムの構成および内容、処理内容、処理順序などは、本発明の趣旨に沿って適宜変更することができる。

【0 0 5 6】

【発明の効果】

本発明によると、J a v aサーバレットによって提供されるW e bアプリケーションの画面のデザインの編集を、J a v aのスキルのない者であっても容易に行うことができるようにすることができる。

【0 0 5 7】

請求項 2 の発明によると、編集の際に画面生成ロジックと業務ロジックとの連携を崩すようなバグを防止することができる。請求項 3 の発明によると、テスト環境から本番環境への移行の際に、U R Lの付替えの手間を少なくすることができる。

【図面の簡単な説明】

【図 1】

本発明に係るW e bサーバが設けられたネットワークシステムの例を示す図である。

【図 2】

W e bサーバのハードウェア構成の例を示す図である。

【図 3】

W e bサーバのプログラムの例を示す図である。

【図 4】

ログイン画面の例を示す図である。

【図 5】

H T M Lのソースコードの例を示す図である。

【図 6】

J a v aソースコードの例を示す図である。

【図 7】

初期状態のW e b ページ編集画面の例を示す図である。

【図 8】

H T M L のソースコードが表示された W e b ページ編集画面の例を示す図である。

【図 9】

W e b サーバの機能的構成の例を示す図である。

【図 1 0】

編集後のログイン画面を示す図である。

【図 1 1】

更新が完了した旨のメッセージの画面を示す図である。

【図 1 2】

画面（W e b ページ）の例を示す図である。

【図 1 3】

図 1 2 の画面のための H T M L のソースコードの例を示す図である。

【図 1 4】

業務ロジックおよび画面生成ロジックにおける X タグの機能を説明する図である。

【図 1 5】

図 1 3 （a）の H T M L のソースコードを編集した例を示す図である。

【図 1 6】

W e b サーバにおける処理の流れを説明するフローチャートである。

【図 1 7】

画面（W e b ページ）の例を示す図である。

【符号の説明】

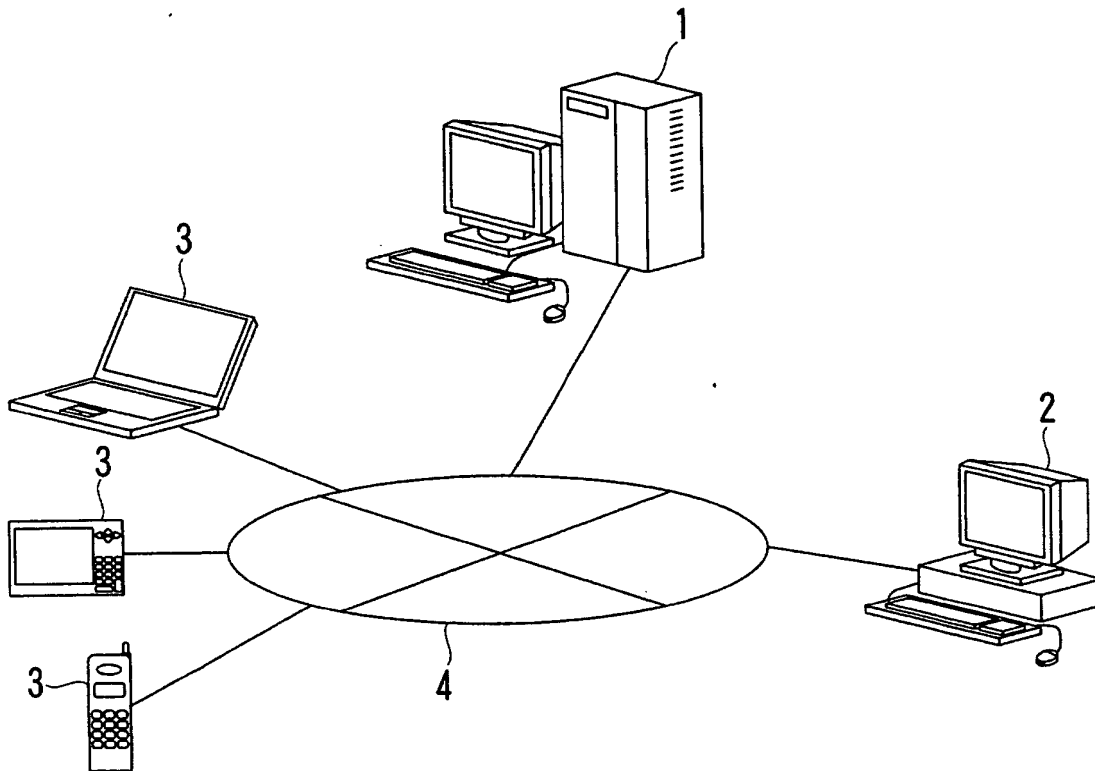
- 1 W e b サーバ
- 2 端末装置
- 8 0 W e b アプリケーション（J a v a プログラム）
- 8 9 パス情報
- 1 0 1 ファイル受信部（取得手段）
- 1 0 3 指定画面抽出部（抽出手段）
- 1 0 4 第一の変換部（第一の変換手段）

- 1 0 5 コード更新部（プログラム更新手段）
- 1 0 6 第二の変換部（第二の変換手段）
- 1 0 7 ファイル送信部（送信手段）
- 1 0 8 パス記憶部（パス記憶手段）
- 1 0 9 パス更新部（パス更新手段）

【書類名】 図面

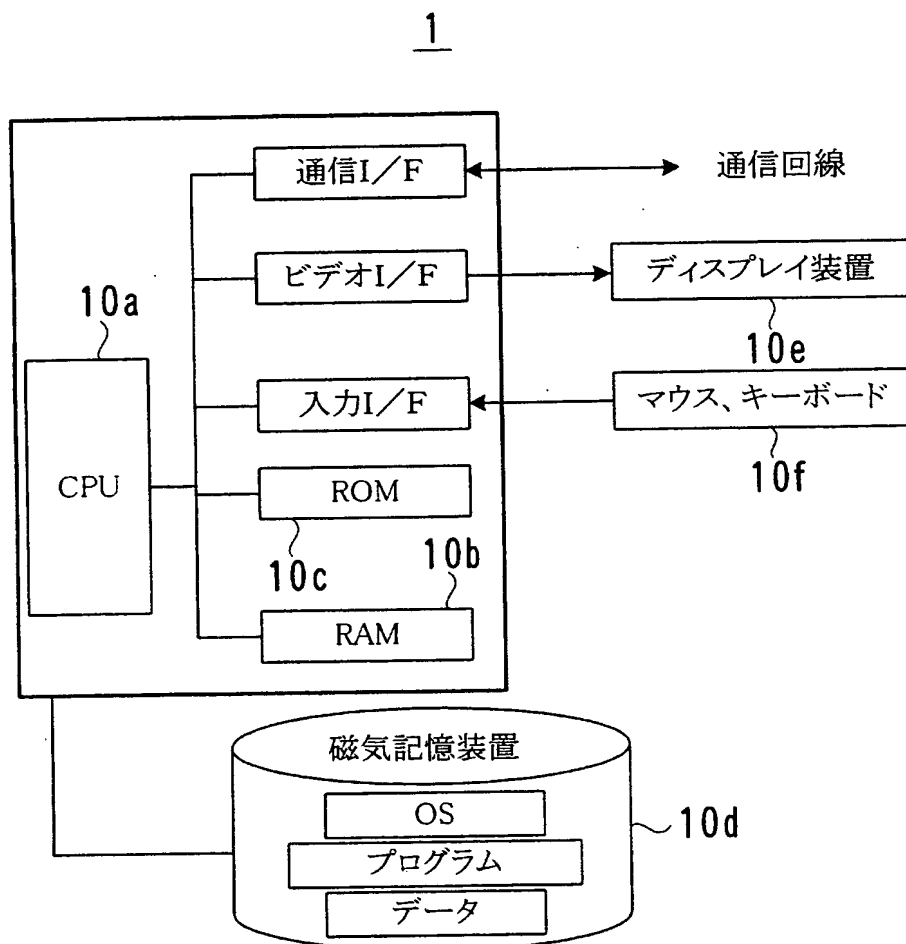
【図 1】

本発明に係るWebサーバが設けられたネットワークシステムの例を示す図



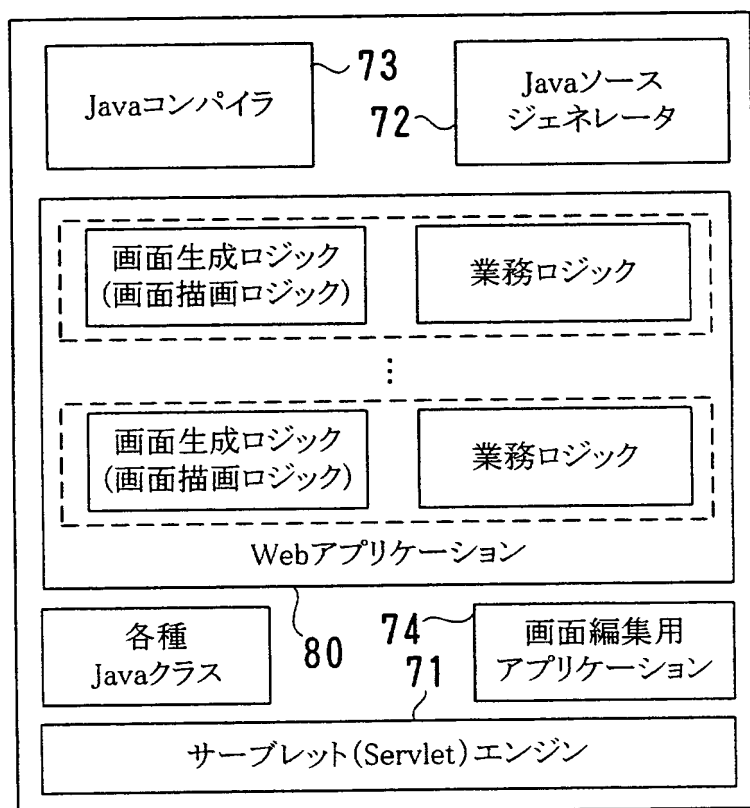
【図 2】

Webサーバのハードウェア構成の例を示す図



【図 3】

Webサーバのプログラムの例を示す図



【図 4】

ログイン画面の例を示す図

Sample Login:GID01

Login(GID01)

User ID

PassWord

OK

HG1

【図 5】

HTMLのソースコードの例を示す図

```
<HTML>
<HEAD>
<TITLE>Sample Login:GID01</TITLE>
<META HTTP-EQUIV="Content-Style-Type" CONTENT="text/css">
<LINK rel="stylesheet" type="text/css" href="test.css">

<X name="JAVASCRIPT_PARAM">
<SCRIPT language="JavaScript">
<!--
var loginError = 0;
//-->
</SCRIPT>
</X>

      中略

</HEAD>
<BODY BGCOLOR="#E0E0F0" onload="Javascript:init()">
<CENTER>
<BR>
<B><H3>富士通Login(GID01)</H3></B>
<BR>
<BR>
<FORM name="form1" method="post" action="App">
  <INPUT type="hidden" name="GID" value="GID01">
  <INPUT type="hidden" name="SCENE" value="1">
  <TABLE>
    <TR>
      <TD>User ID</TD>
      <TD><INPUT type="text" name="USER_ID" value="X=USER_ID=X"></TD>
    </TR>
    <TR>
      <TD>PassWord</TD>
      <TD><input type="password" name="PassWord"></TD>
    </TR>
    <TR>
      <TD></TD>
      <TD align="right">
        <input type="button" name="sub" value=" OK " disabled="true" onclick="javascript:goNext();">
      </TD>
    </TR>
  </TABLE>
</FORM>
</CENTER>
<BR>
</BODY>
</HTML>
```

【図 6】

Javaソースコードの例を示す図

```

package xx.xx.fks.sample;

import java.io.*;
import java.lang.*;
import java.util.*;
import xx.xx.fks.jaf.core.*;

public class GID01Document extends Document
{
    public void init() {
        protected void _editDocHeader()
        {
            中略
            _editLine("<BR>");
            _editLine("<B><H3>Login(GID01)</H3></B>");
            _editLine("<BR>");
            _editLine("<BR>");
            _editLine("<FORM name="+_form1+" method="+_post+" action="+_getActionPath0 + "App"+_get">");
            _editLine("    <INPUT type="+_hidden+" name="+_GID+" value="+_GID01+">");
            _editLine("    <INPUT type="+_hidden+" name="+_SCENE+" value="+_I+">");
            _editLine("    <TABLE>");
            _editLine("        <TR>");
            _editLine("            <TD>");
            _editLine("                User ID");
            _editLine("            </TD>");
            _editLine("            <TD>");
            _editLine("                <INPUT type="+_text+" name="+_USERID+" value="+_getHTML("USERID")+_get">");
            中略
            _editLine("</HTML>");
        }
    }
}

```

【図 7】

初期状態のWebページ編集画面の例を示す図

[Document Maintenance]

BX21 BX22 BX23 BN21 BN22

GID User ID Pass Word

BX24

HG2

【図 8】

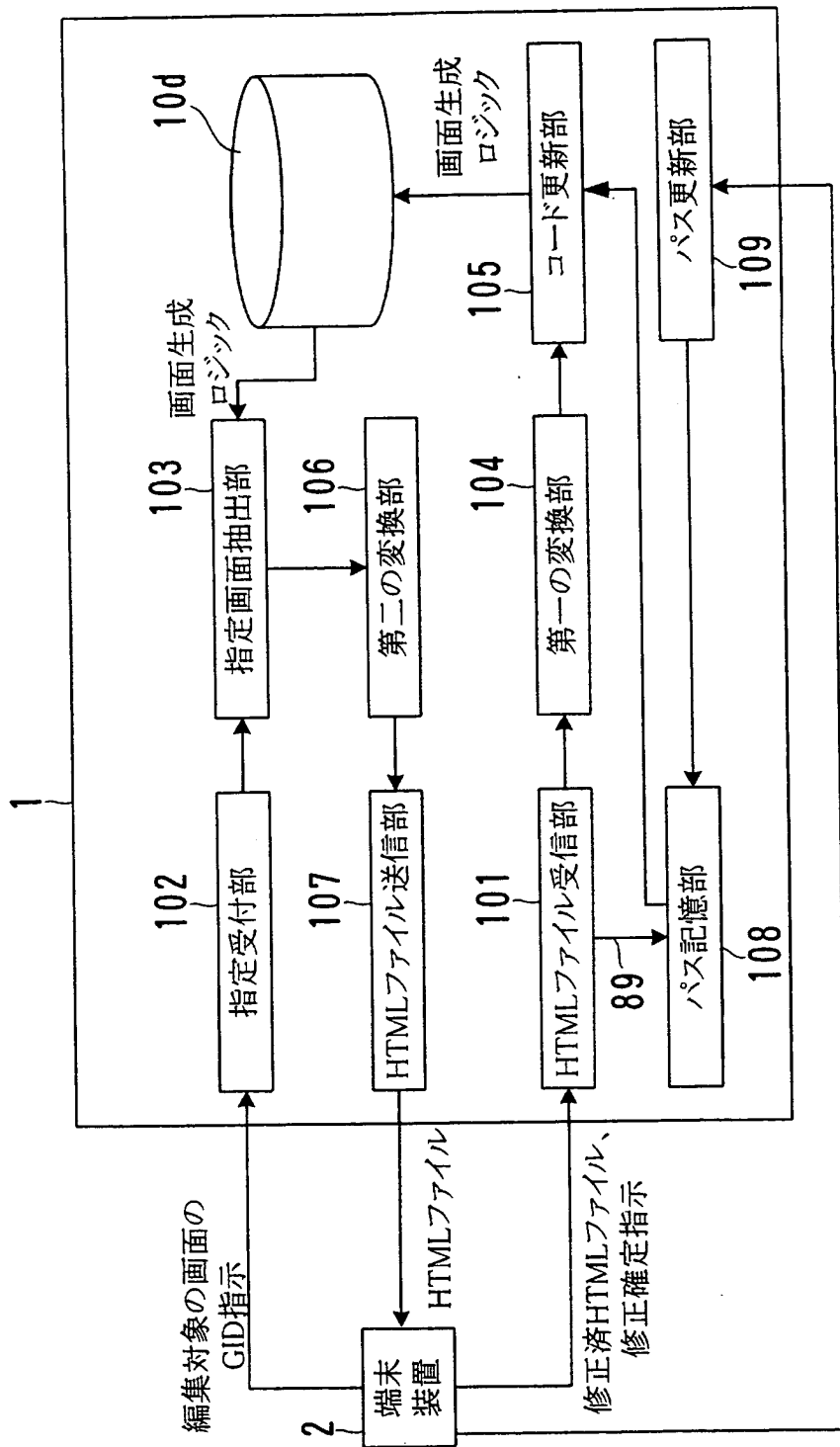
HTMLのソースコードが表示されたWebページ編集画面の例を示す図

The screenshot shows a web browser window titled "[Document Maintenance]". At the top, there are five labels: BX21, BX22, BX23, BN21, and BN22. Below these are input fields for "GID" (containing "GID01"), "User ID", and "Pass Word", followed by two buttons: "Design Edit" (labeled BN21) and "Update Design" (labeled BN22). The main area of the browser displays HTML source code. The code includes a JavaScript function for login error handling, a "goNext" function, and "buttonDisable" and "buttonEnable" functions. It also contains HTML tags for the body background color and onload event. At the bottom of the browser window, there is a "Preview HTML" button (labeled BN23) and a label BX24. The browser window is labeled HG2 at the bottom right.

```
if(loginError == 1) {  
    alert("Login false.Please check UserID and  
    PassWord.");  
}  
buttonEnable();  
document.forms.form1.USER_ID.focus();  
}  
  
function goNext(){  
    buttonDisable()  
    document.forms.form1.submit();  
}  
  
function buttonDisable(){  
    document.forms.form1.sub.disabled=true;  
}  
function buttonEnable(){  
    document.forms.form1.sub.disabled=false;  
}  
  
//---->  
</SCRIPT>  
</HEAD>  
<BODY BGCOLOR="#E0E0F0" onload="Javascript:init()">  
<CENTER>  
<BR>
```

【図 9】

Webサーバの機能的構成の例を示す図



【図 10】

編集後のログイン画面を示す図

Sample Login:GID01

ログイン画面 (GID01)

システム稼動時間 7:00 ~ 22:00

ユーザID

パスワード

OK

〇×株式会社

HG1'

【図 11】

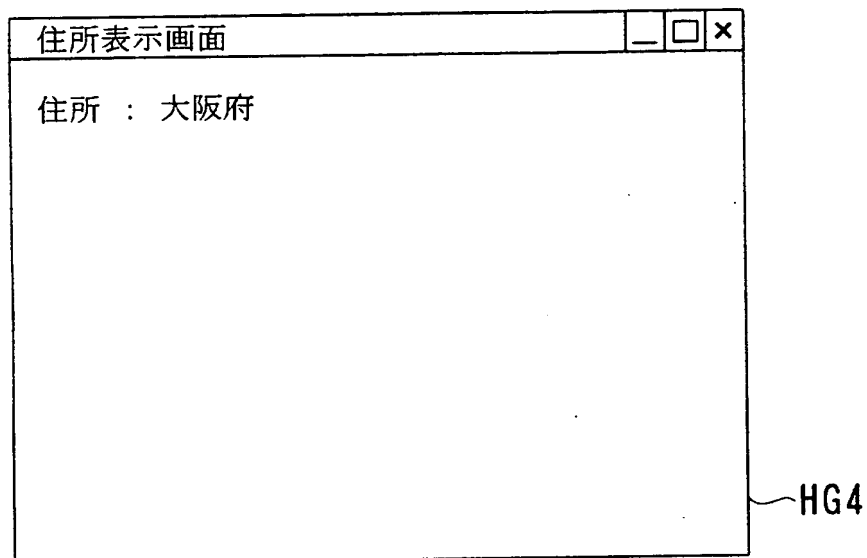
更新が完了した旨のメッセージの画面を示す図

Update Done

OK

【図 1 2】

画面 (Web ページ) の例を示す図



【図 13】

図12の画面のためのHTMLのソースコードの例を示す図

(a)

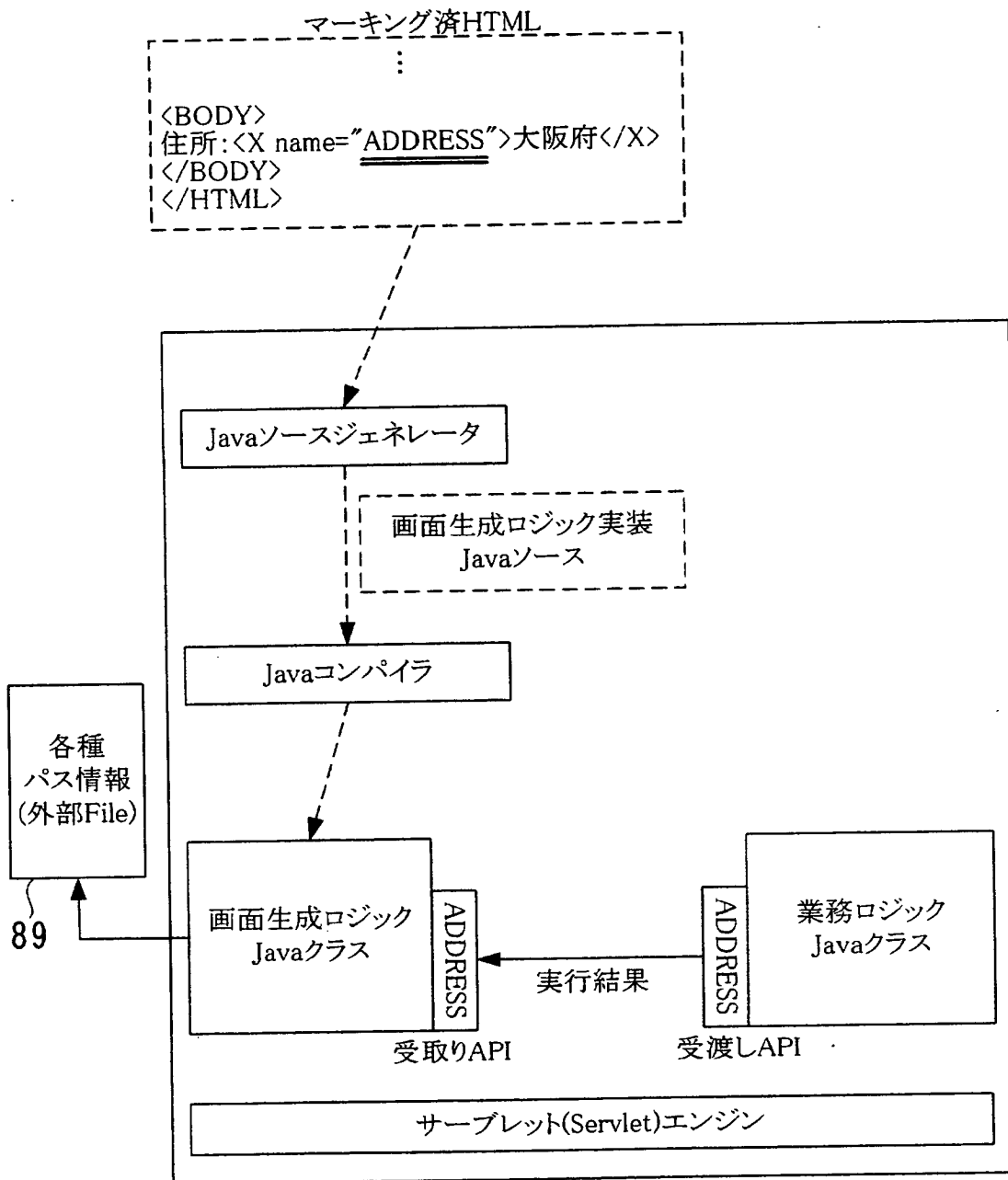
```
<HTML>
<HEAD><TITLE>
住所表示画面
</TITLE></HEAD>
<BODY>
住所:大阪府
</BODY>
</HTML>
```

(b)

```
<HTML>
<HEAD><TITLE>
住所表示画面
</TITLE></HEAD>
<BODY>
住所:<X name="ADDRESS">大阪府</X>
</BODY>
</HTML>
```

【図 14】

業務ロジックおよび画面生成ロジックにおけるXタグの機能を説明する図



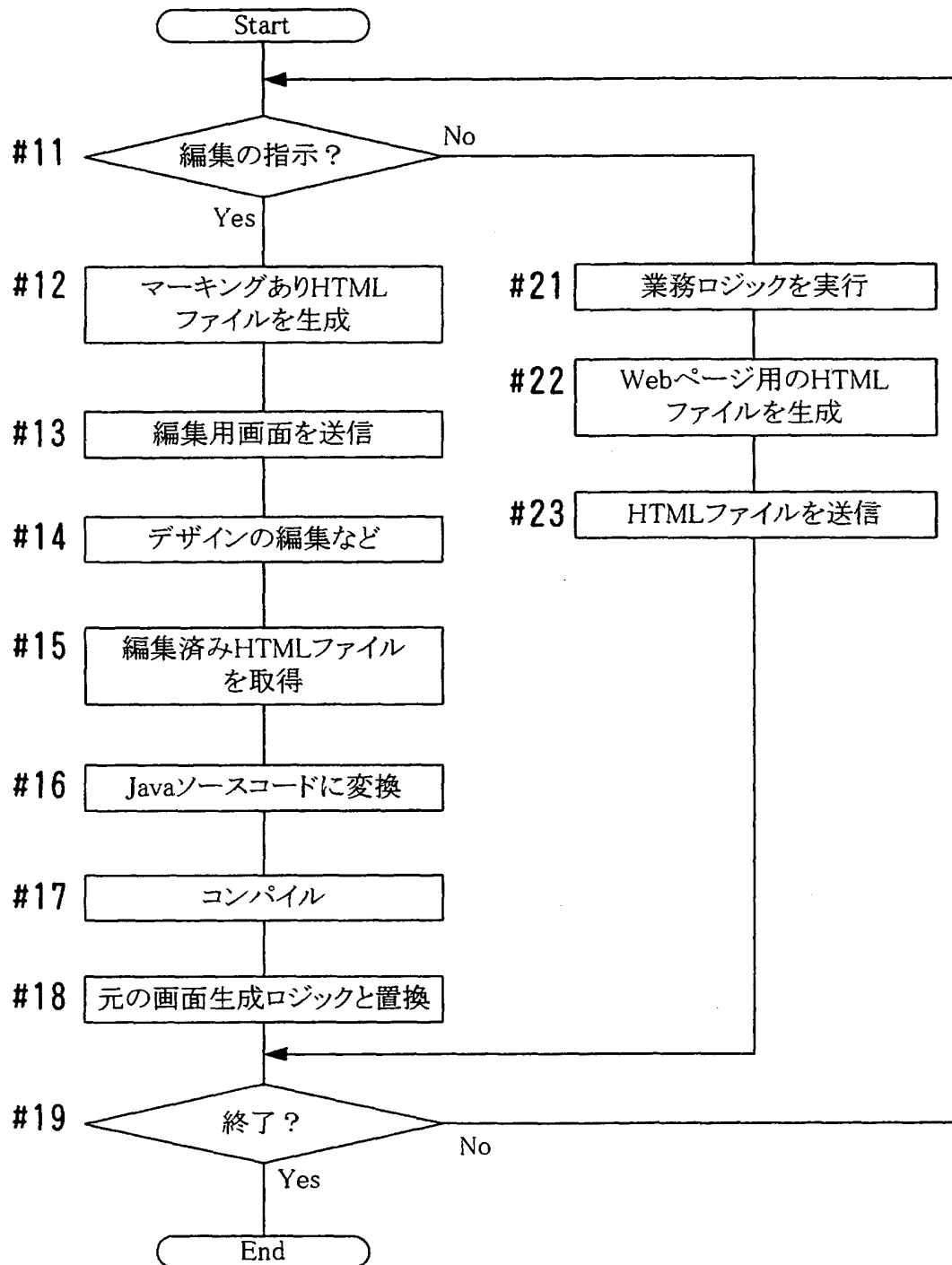
【図 15】

図13(a)のHTMLのソースコードを編集した例を示す図

```
<HTML>
<HEAD><TITLE>
住所表示画面
</TITLE></HEAD>
<BODY BGCOLOR="#AAEEBB">
<CENTER>
<TABLE border="1"><TR><TD>
<FONT size="6">
★ ユーザ情報 ★
</FONT>
</TD></TR></TABLE>
</CENTER><BR>
あなたの住所は以下のとおりです。<BR>
ご確認ください。<BR>
<HR color="#0000FF">
<FONT size="8"><CENTER>
<X name="ADDRESS">大阪府</X>
</CENTER></FONT>
<HR color="#0000FF">
<TABLE width="100%">
<TR><TD align="right">
<IMG src="fujitsu.gif" height="50"><BR><BR>
<FONT size="2"><B>2002 富士通株式会社</B></FONT>
</TD></TR>
</TABLE>
</BODY>
</HTML>
```

【図 16】

Webサーバにおける処理の流れを説明するフローチャート



【図 17】

画面 (Web ページ) の例を示す図

住所表示画面

★ ユーザ情報 ★

あなたの住所は以下のとおりです。
ご確認ください。

大阪市中心区域見

○×株式会社

MV

【書類名】 要約書

【要約】

【課題】 J a v a サブレット環境下で容易に W e b ページの編集を行う。

【解決手段】 W e b ページの HTML ファイルを生成する J a v a のプログラムを記憶する W e b サーバ 1 に、端末装置 2 から HTML ファイルを取得する HTML ファイル受信部 1 0 1 と、取得した HTML ファイルを J a v a のコード群に変換する第一の変換部 1 0 4 と、変換した J a v a のコード群をプログラムのうちの W e b ページの構成に係る部分と置き換えてプログラムを更新するコード更新部 1 0 5 と、端末装置 2 が指定した W e b ページの構成に係る部分の J a v a のコード群をプログラムの中から抽出する指定画面抽出部 1 0 3 と、抽出した J a v a のコード群を HTML ファイルに変換する第二の変換部 1 0 6 と、変換した HTML ファイルを端末装置 2 に送信する HTML ファイル送信部 1 0 7 と、を設け、第一の変換部 1 0 4 は、取得した HTML ファイルに修正不可の部分を示す X タグが含まれている場合に、X タグを、修正不可の部分を示す J a v a のコードに変換し、第二の変換部 1 0 6 は、抽出された J a v a のコード群に含まれる修正不可の部分を示すコードを X タグに変換する。

【選択図】 図 9

特願 2 0 0 2 - 2 4 9 2 6 6

出 願 人 履 歴 情 報

識別番号

[0 0 0 0 0 5 2 2 3]

- | | |
|----------|-----------------------------|
| 1. 変更年月日 | 1 9 9 0 年 8 月 2 4 日 |
| [変更理由] | 新規登録 |
| 住 所 | 神奈川県川崎市中原区上小田中 1 0 1 5 番地 |
| 氏 名 | 富士通株式会社 |
| | |
| 2. 変更年月日 | 1 9 9 6 年 3 月 2 6 日 |
| [変更理由] | 住所変更 |
| 住 所 | 神奈川県川崎市中原区上小田中 4 丁目 1 番 1 号 |
| 氏 名 | 富士通株式会社 |